KISS-BOX QCONDUCTOR/M

EZ-CC MODULES FAMILY

USER'S MANUAL

Contents

1 - Introduction....3

2 - Installation....3

2.1 - Mac installation....3

2.2 - Windows installation....3

3 - Integration in host....4

3.1 - Using EZ-CC modules in Max/MSP....4

3.1.1 - Using EZ-CC modules in a Max/MSP patch....4

3.1.2 - Controlling EZ-CC modules from Max/MSP....5

3.1.2.1 - Recalling a cue....5

3.1.2.2 - Saving and restoring cue banks....6

3.1.2.3 - Getting data from Kiss-Boxes....6

3.2 - Using EZ-CC modules with other VST hosts....7

4 - Modules reference....8

4.1.1 - EZ-CC DMX....8

4.1.1.1 - Configuration process....8

4.1.1.2 - Programming cues....9

4.1.1.3 - Controlling DMX channels from host....10

4.1.2 - EZ-CC I/O....11

4.1.2.1 - Configuration process....11

4.1.2.2 - Programming cues....12

4.1.2.3 - Controlling output channels from host....12

4.1.3 - EZ-CC MIDI....13

4.1.3.1 - Configuration process....13

4.1.3.2 - Programming cues....14

4.1.3.3 - Controlling MIDI output stream from host....14

4.1.3.4 - Sending MIDI input stream to host....14

4.1.4 - EZ-CC Serial4....15

4.1.4.1 - Configuration process....15

4.1.4.2 - Programming cues....15

4.1.4.3 - Creating and sending strings from host....16

5 - Document revisions....17

1 - Introduction

The QConductor/M EZ-CC modules belong to Kiss-Box Show Control software products family are the easiest one to use. These modules have been optimized for integration in Max/MSP host, but since they take the form of VST plugins, they can be integrated in any VST host.

These modules are mainly output modules, used to command external devices, but some of them are able to read incoming data from external devices. They do not require the QConductor Manager module to run, thus they can not receive any message coming from another module.

All EZ-CC members share the following characteristics:

- direct control of one physical interface (typically a single Kiss-Box or a port on a Kiss-Box)
- 128 cues stored in the module memory and available as "banks" and "presets"
- Each cue is seen as a VST preset, than can be saved/restored from the host using standard VST files
- Cues are recalled using Program Change MIDI message
- User Interface for direct control and cue programming
- Available both on Windows (32 bits) and Mac OS-X platforms

2 - Installation

2.1 - Mac installation

Kiss-Box provides an installation package, compatible with OSX versions 10.4, 10.5 and 10.6. Double-click on the installation package file and follow the instructions given on the different screens.

The installer will copy the modules in the computer's VST directory, along with the other installed plugins. We highly recommend that you do not change the destination folder, since all VST compatible applications will look in this folder to find installed plugins. Placing the EZ-CC modules in a different folder will make them unreachable from Max/MSP and most other VST hosts.

2.2 - Windows installation

Kiss-Box provides an installer application, compatible with any 32 bits version of Windows from XP. Start the setup application and follow the instructions given in the different screens.

Note that there is not really a standard folder for VST plugins on the Windows platform. Most applications look for **/Program Files/steinberg/vstplugins** folder by default. However, this default folder can be changed at any moment, so it is possible that the plugins are located in a different folder on some computers.

The installer offers you to install the EZ-CC modules in that folder by default, but you can change it during installation if necessary.

3 - Integration in host

3.1 - Using EZ-CC modules in Max/MSP

IMPORTANT NOTES:

- EZ-CC modules have been specifically developed and tested for Max 5. Kiss-Box does not guarantee the compatibility with previous versions of Max/MSP
- Since EZ-CC modules are VST plugins, they require MSP extension to run

3.1.1 - Using EZ-CC modules in a Max/MSP patch

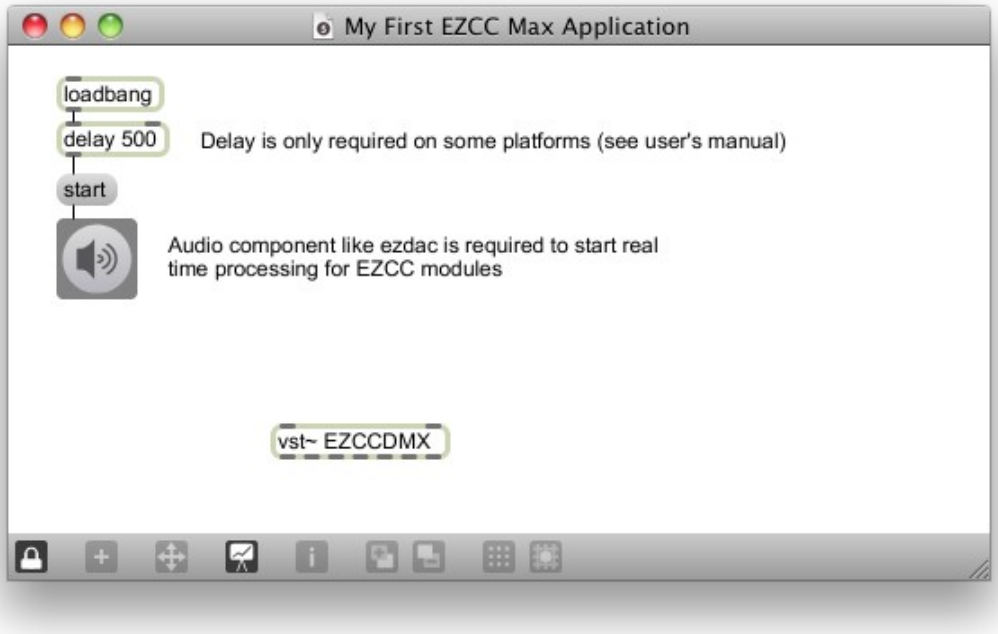
Start Max/MSP and open a new Patcher window (File / New Patcher)

Double-click or type Ctrl+Shift+O to create a new object and choose "Object" in the list that appears. A empty object is then created on the Patcher workspace.

Enter "vst~" (without the quotes) in the Object box, followed by the name of the EZ-CC module you want to use (e.g: vst~ KBIOEZCC for the I/O EZ-CC module). Click outside of the vst~ object to load the EZ-CC module (the enter key would create a new line in the object).

Starting from now, you can already use your module and create your first cues (and you understand why this family is named EZ-CC... because it's really easy to use compared to many other show-control applications). However, you must keep in mind that Max/MSP "freezes" all VST modules until the audio engine is started. Your Patch must then contain an audio object that activates the audio engine. This audio object can be placed anywhere in your project (even on a different subpatch), but **it must exist**.

The following screen-shot shows an example of automatic activation of the audio engine when a patch is loaded, based on an ezdac~object (other audio objects like ezadc~ can also be used. Please refer to Max/MSP user's guide).


My First EZCC Max Application
loadbang
delay 500
Delay is only required on some platforms (see user's manual)
start
Audio component like ezdac is required to start real time processing for EZCC modules
vst~ EZCCDMX

Note that you do not need to connect any audio input or output the a EZ-CC module. All EZ-CC modules provide a stereo pass-through mechanism (the audio signals will pass through the modules without being affected in any way).

IMPORTANT NOTICE: on some platforms, an activation sequence problem can occur with the EZ-CC modules, depending on the position of the different Max objects. In that case, the EZ-CC modules receives a suspend instruction from Max and becomes inactive. The easiest way to solve this problem is to delay slightly the *start* message when the patch is started. The previous screen-shot shows you how to insert the *delay* object on the message path to *ezdac*.

Do not forget that you need to lock the Patch before you can access modules and use them if you double-click on the EZ-CC object. You can also add a *open* message to the first EZ-CC inlet and click on it.

3.1.2 - Controlling EZ-CC modules from Max/MSP

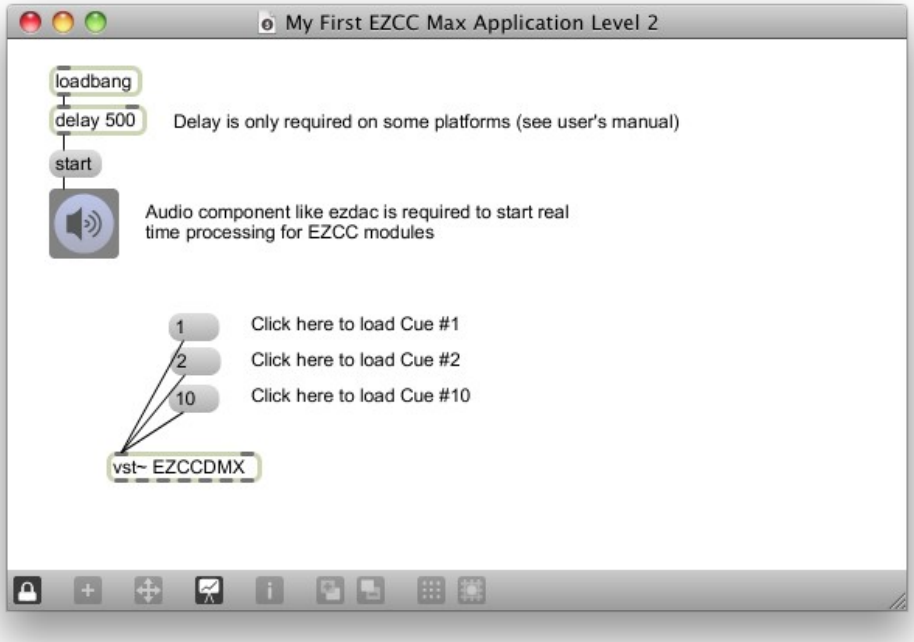
You have to keep in mind that Max differs from many other VST hosts by the fact that you have a full control over all messages sent to the different modules. That means that Max does not connect anything "by default" to your module to control and lets you decide what you want to send (and consequently do) to the modules.

In particular, Max/MSP does not perform "default" operations (like loading or saving presets/banks) when you load or close your Patch. The following sections will explain you how to implement basic control functions on EZ-CC modules in Max.

3.1.2.1 - Recalling a cue

All EZ-CC modules use MIDI Program Change message to recall pre-programmed cues. You can them control directly the modules using any MIDI object on Max.

It is also possible to call a cue directly by sending its number on the leftmost inlet of the EZ-CC module. Note that cue numbers are between 1 and 128 while Program Change values are between 0 and 127.

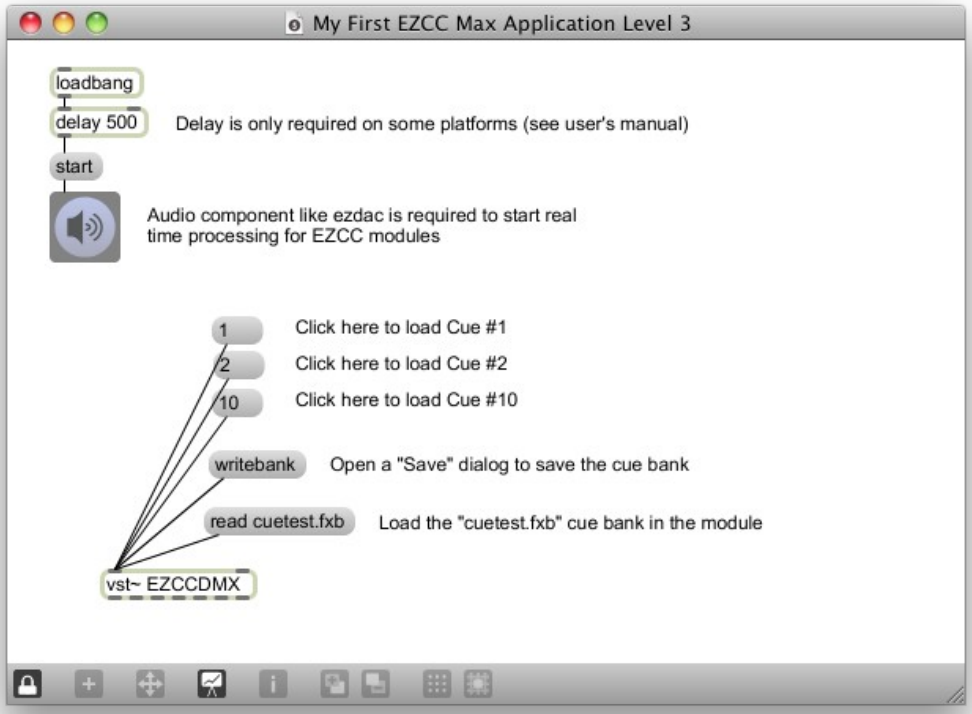

My First EZCC Max Application Level 2
loadbang
delay 500
Delay is only required on some platforms (see user's manual)
start
Audio component like ezdac is required to start real time processing for EZCC modules
1
Click here to load Cue #1
2
Click here to load Cue #2
10
Click here to load Cue #10
vst~ EZCCDMX

3.1.2.2 - Saving and restoring cue banks

Unlike many other hosts, Max/MSP lets you choose when cues shall be loaded/save from/to disk. It must be noticed that Max **never** saves automatically banks.

In order to save a cue bank from an EZ-CC module, you shall use the *writebank* message, sent on the first inlet of the EZ-CC module. If you do not specify a file name after *writebank*, Max will open a standard Save dialog, which allows you to choose the filename for the bank. If a file name is specified in the message, the file is saved under the given name when the message is sent to the module.

The *write* message is similar to *writemessage*, but it saves only the current cue.


My First EZCC Max Application Level 3
loadbang
delay 500
Delay is only required on some platforms (see user's manual)
start
Audio component like ezdac is required to start real time processing for EZCC modules
1
Click here to load Cue #1
2
Click here to load Cue #2
10
Click here to load Cue #10
writebank
Open a "Save" dialog to save the cue bank
read cuetest.fxb
Load the "cuetest.fxb" cue bank in the module
vst~ EZCCDMX

You can load a cue bank or a single cue using the *read* message. Max will automatically select the correct file type (bank or single cue) depending on the file content.

3.1.2.3 - Getting data from Kiss-Boxes

Some EZ-CC modules are also able to provide data read from the Kiss-Boxes to the Max patch. It is then possible to make the patch react to inputs coming from the Kiss-Box.

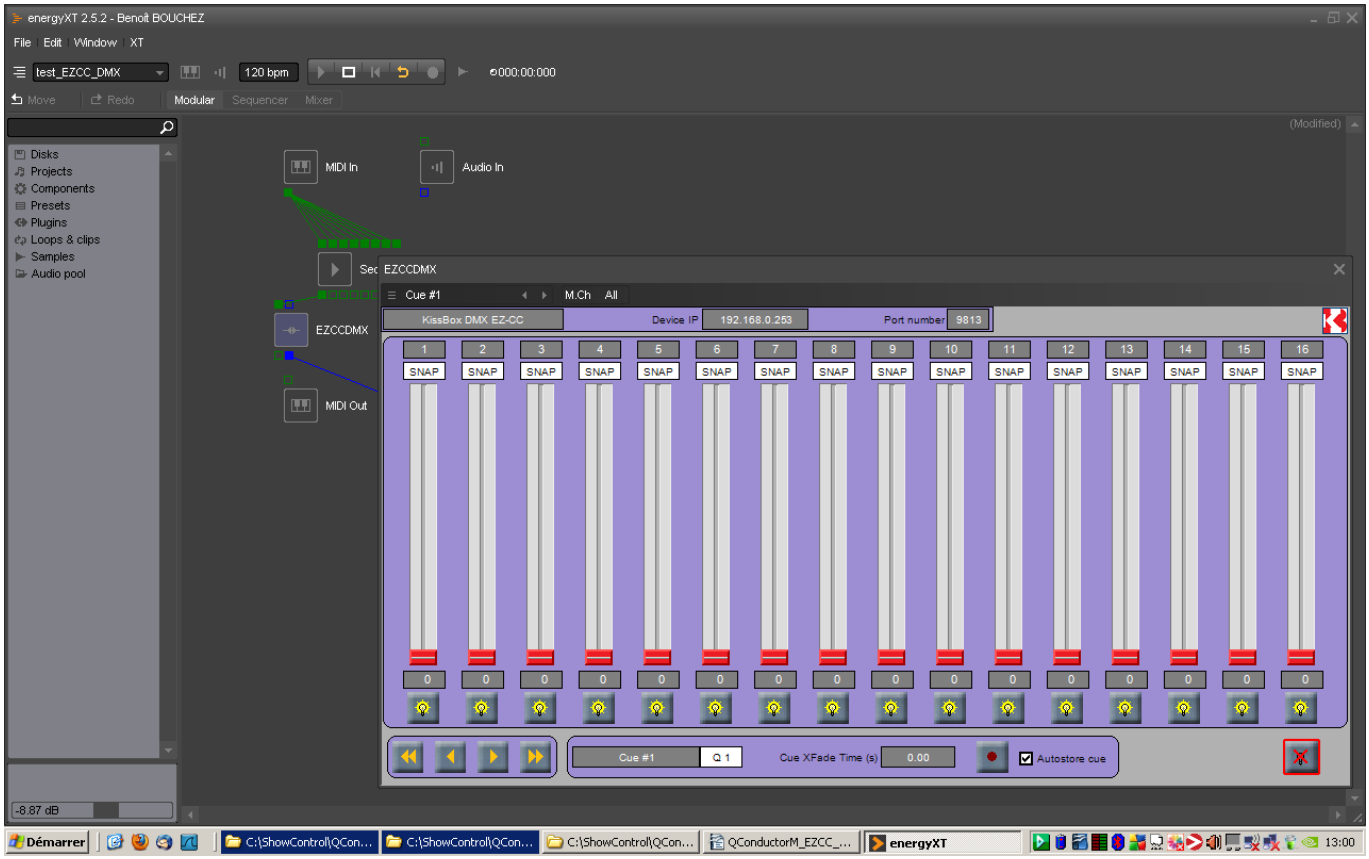
The data are available in the form of MIDI messages, provided by the EZ-CC module on the patch. Due to this limitation, only some EZ-CC modules are able to provide input data to the patch (see details for each EZ-CC module in the next pages)

MIDI messages are available on the fifth outlet. Please report to the module description for the description of the MIDI data provided by each module.

Note: if a patch needs to retrieve native data coming from any Kiss-Box, you may need to use the QConductor/M Manager based modules.

3.2 - Using EZ-CC modules with other VST hosts

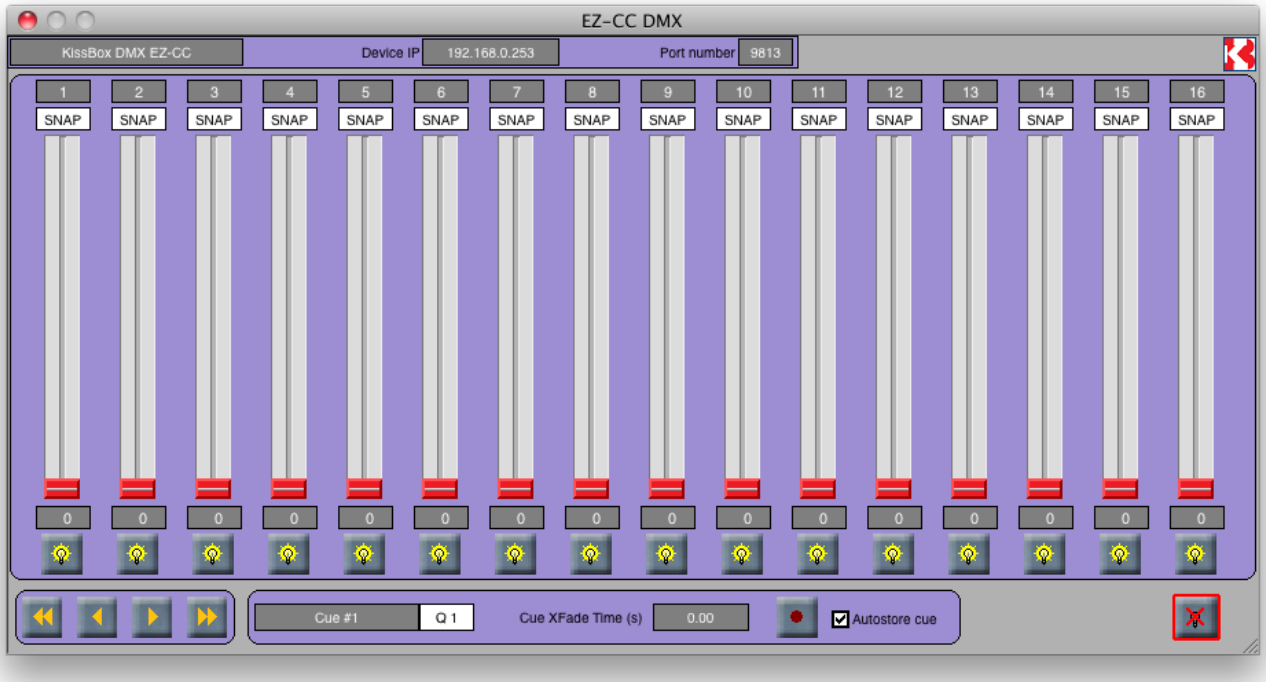
The EZ-CC modules are optimized designed to be used in Max/MSP programming environment. But since they are VST2.4 compliant plugins, they can also be used in any other VST-compatible host. However, some functionality may not be available in that case, since many hosts do not provide signal routing capabilities as Max/MSP does.



4 - Modules reference

4.1.1 - EZ-CC DMX

The EZ-CC DMX module is designed to control one DMX Universe output. It stores 128 cues, that are programmed using the integrated graphical user interface. The host can override stored DMX values at any moment, using MIDI or VST values. It is also possible to control dynamically any DMX channel using the graphical user interface or a MIDI surface control via VST automation.



4.1.1.1 - Configuration process

Each EZ-CC DMX module uses a default configuration when it is instantiated in the host. You need to change this configuration before the EZ-CC is able to control the remote Kiss-Box.

The top leftmost edit box allows the user to enter a name for the module, in order to help the programmer to remember what Kiss-Box is controlled by the module (e.g: "Stage Left DMX group").

Enter in *Device IP* edit box the IP address of the DMX Kiss-Box you want to control from this module. Then enter in *Port Number* box the UDP port number used by the DMX Kiss-Box (this value is defined with the Kiss-Box Editor).

IMPORTANT: the Kiss-Box controlled by the EZ-CC DMX modules must be configured with the Kiss-Box Editor in UDP mode.

The module configuration is stored in the cue bank. You need to save the bank in order to save the configuration parameters (see section 3.1.2.2).

4.1.1.2 - Programming cues

The EZ-CC DMX module stores up to 128 cues. Each of these cues contain the configuration for the 512 DMX channels. When a cue is recalled, each DMX channel receives the value programmed for this cue, but each channel can have a different behavior.

You can browse the 512 channels using the arrow buttons located on the left bottom corner of the GUI. The dual arrow buttons are used to browse the channels by page of 16 channels. The single arrow buttons browse one channel at a time. The channel numbers are displayed on the top row, just under the module configuration area.

For each channel in a cue, the user can choose between 3 modes, using the selector located under the channel number:

- OUT Q: the DMX channel is not updated when the cue is loaded in the module. The DMX channel remains at the last received value.
- SNAP: the DMX channel is updated immediately when the cue is loaded. The DMX value sent on the channel is defined using the slider or incoming MIDI control change (see below)
- FADE: the DMX channel will fade from the current value to the value defined during programming. The fade time for the cue is defined using the *Cue XFade Time* edition box.

Note that all channels are using LTP (Latest Takes Precedence) logic. Any MIDI/VST message coming from the host or from a slider overrides the DMX value stored in the cue.

The channel sliders offer two resolutions. If you just drag a channel slider with the mouse, you will be able to set a coarse DMX value. If you need an higher precision, hold the shift key on the keyboard while you move the slider.

IMPORTANT: the Flash and Blackout buttons states **are not** saved in the cues.

Each cue can be renamed using the edition box near the channel browsing buttons.

Values and modes of a cue can be stored in memory automatically or not, depending on the **Autostore cue** checkbox. When this box is checked, each action made on a channel slider or channel mode selector is saved automatically.

If you do not want to save the changes automatically, uncheck the **Autostore cue** checkbox. You will then need to press the **Record** button (button with the red dot on the bottom line) to store the cue parameters in memory.

Note that these buttons do not store cues on disk, but only in memory. See section 3.1.2.2 for the details of cue storage on mass memory.

4.1.1.3 - Controlling DMX channels from host

The DMX EZ-CC module can be controlled by MIDI and by VST parameters.

There are 512 VST parameters defined in the module, each matching a DMX channel. VST values (0.0 to 1.0) have a linear relationship to DMX values (0 to 100%).

In case of MIDI control, the EZ-CC DMX recognizes Control Change and Note On / Note Off messages. Since there are only 128 controllers/notes per MIDI channel, the module uses the channels 1 to 4.

MIDI Channel	Note / Control	DMX Channel
1	0..127	1..128
2	0..127	129..256
3	0..127	257..384
4	0..127	385..512

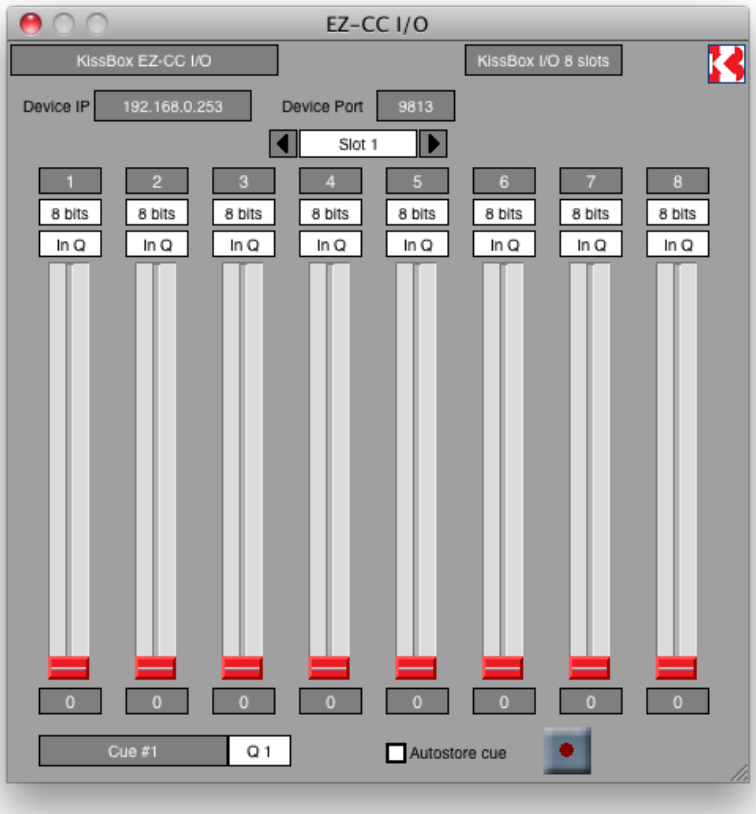
The Control Change value and the Note velocity value are used to compute the DMX value. Velocities/control values 0 to 127 are transformed to DMX values 0 to 254.

4.1.2 - EZ-CC I/O

The EZ-CC I/O module is designed to control the following Kiss-Box models:

- Kiss-Box 8 slots I/O
- Kiss-Box 3 slots I/O
- PowerSwitcher4
- PowerSwitcher6

It stores 128 cues, that are programmed using the integrated graphical user interface. The host can control any output channel at any moment, using MIDI or VST values.



4.1.2.1 - Configuration process

Each EZ-CC I/O module uses a default configuration when it is instantiated in the host. You need to change this configuration before the EZ-CC is able to control the remote Kiss-Box.

IMPORTANT NOTE: the I/O Kiss-Box shall be configured in UDP mode. Please refer to Kiss-Box Editor User's Manual.

The top leftmost edit box allows the user to enter a name for the module, in order to help the programmer to remember what Kiss-Box is controlled by the module (e.g: "Museum entry desk").

Enter in *Device IP* edit box the IP address of the I/O Kiss-Box you want to control from this module. The *UDP Port* edition box shall be filled with the UDP port number declared during the Kiss-Box configuration process (see Kiss-Box Editor User's Manual).

Select also the model of the Kiss-Box I/O system that will be controller from the module using the right-most selection box.

To complete the configuration process, you have to specify for each channel of each slot the data format expected by the I/O card installed in the slot. This is done by the selection list located under the channel number. Use **O/1** mode for digital I/O (like DI8DC or DO4PR) and **8 bits** mode for analog I/O.

NOTE: the complete module configuration is stored in the cue bank. You need to save the bank in order to save the configuration parameters (see section 3.1.2.2).

4.1.2.2 - Programming cues

The KB-I/O-EZCC module stores 128 cues. Each of these cues contain the configuration for the 8 slots that can exist in a Kiss-Box I/O card-cage. When a cue is recalled, each I/O channel will receive the value programmed for this cue, but each channel can have a different behavior.

You can browse the different slots using the arrow buttons located on both sides of the slot indicator. The number of slots that you can browse depends on the Kiss-Box model chosen during configuration step.

For each channel of a slot, the user can choose if the channel is updated with a new value (**In Q**) or shall keep its current value (**Out Q**).

Note that all channels are using LTP (Latest Takes Precedence) logic. Any MIDI/VST message coming from the host or from a slider overrides the channel value stored in the cue.

Each cue can be renamed using the edition box at bottom of the GUI window.

Values and modes of a cue can be stored in memory automatically or not, depending on the **Autostore cue** checkbox. When this box is checked, each action made on a channel slider or channel mode selector is saved automatically.

If you do not want to save the changes automatically, uncheck the **Autostore cue** checkbox. You will then need to press the **Record** button (button with the red dot on the bottom line) to store the cue parameters in memory.

4.1.2.3 - Controlling output channels from host

The EZ-CC I/O module can be controlled by MIDI and by VST parameters.

There are 64 VST parameters defined in the module, defined as 8 groups (one per slot) of 8 channels. If the channel is configured as boolean, a VST value lower than 0.5 deactivates the output channel, and a value greater than 0.5 activates the output channel. Analog channels can be controlled linearly by VST values from 0.0 to 1.0.

In case of MIDI control, the KB-IO-EZCC recognizes Control Change and Note On / Note Off messages. The module only listens on MIDI channel 1.

Digital outputs are reset when CC value or Note velocity is lower or equal to 63. Value larger or equal to 64 sets the digital output.

Analog outputs are linearly controlled by CC value or Note velocity from 0 to 127, translated to 8 bits value 0 to 254.

Slot	CC Number / Note Number
1	0 to 7
2	8 to 15
3	16 to 23
4	24 to 31
5	32 to 39
6	40 to 47
7	48 to 55
8	56 to 63

4.1.3 - EZ-CC MIDI

The EZ-CC MIDI is designed to control a single RTP-MIDI Kiss-Box port. It can be used to control the following Kiss-Box models:

- RTP-MIDI (2 ports)
- LTC-MIDI (LTC port and MIDI port)
- HIN Box

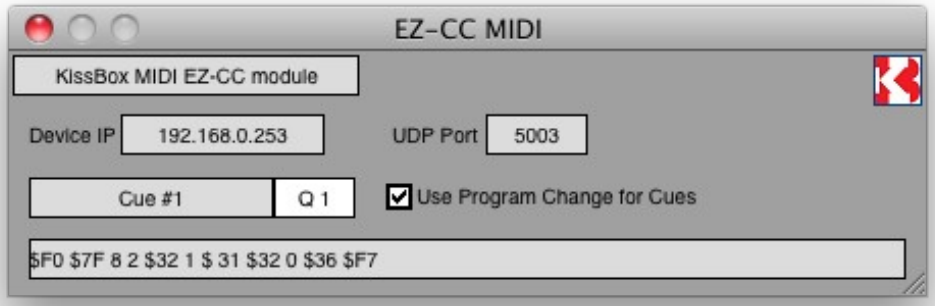
The RTP-MIDI Kiss-Box provides two physical MIDI ports. Two KB-MIDI-EZCC modules are then required to control the two ports.

The MIDI port of the LTC-MIDI Kiss-Box can also be controlled from a single module. The LTC port of this model of Kiss-Box can also be controlled from the module, but only MTC message can be transmitted to the Kiss-Box (other MIDI messages are ignored by the LTC port)

EZ-CC MIDI module allows to send any existing MIDI message (including any form of System Exclusive, like MIDI Show Control or MIDI Machine Control) without dealing with a complex patch to produce the message data. It is also possible to combine multiple MIDI messages in the same cue.

MIDI message produced on the Max/MSP patch received by the module will be transmitted to the MIDI port of the Kiss-Box (excluding System Exclusive).

A user can then combine easily both kind of message to simplify the MIDI programming.



4.1.3.1 - Configuration process

Each KB-MIDI-EZCC module uses a default configuration when it is instantiated in the host. You need to change this configuration before the EZ-CC is able to control the remote Kiss-Box.

IMPORTANT NOTE: the MIDI Kiss-Box must be configured for "Peer to Peer" mode. Please refer to Kiss-Box Editor User's Manual to know how to set the Kiss-Box profile.

The top leftmost edit box allows the user to enter a name for the module, in order to help the programmer to remember what Kiss-Box is controlled by the module (e.g: "Audio console C").

Enter in *Device IP* edit box the IP address of the MIDI Kiss-Box you want to control from this module. The *UDP Port* edition box shall be filled with the UDP **control** port number declared during the Kiss-Box configuration process (see Kiss-Box Editor User's Manual). Please take care about the port number that shall be used specifically on Kiss-Box products. The **data** port number is computed automatically from the control port number.

The "Use Program Change for Cues" allows you to select if Program Change messages received from the

host will be transmitted directly to MIDI OUT (left the check-box unchecked) or if the Program Change command will trigger a new cue and send the related MIDI data (check the check-box).

NOTE: the complete module configuration is stored in the cue bank. You need to save the bank in order to save the configuration parameters (see section 3.1.2.2).

4.1.3.2 - Programming cues

The EZ-CC MIDI module stores 128 cues. Each of these cues contains a MIDI message (or a set of MIDI messages) that will be transmitted to MIDI OUT port when the cue is activated. The maximum size of a MIDI message set is 64 bytes.

Each cue can be renamed using the edition box on the GUI window.

Note that cue parameters are saved automatically if a new cue is called. You can then switch from a cue to another without needing to save the changes each time.

The MIDI message can be entered in the cue editor using three different formats:

- hexadecimal (enter the MIDI code in hexadecimal with a "$" in front of the code)
- decimal (enter the MIDI code directly in the string)
- an ASCII string (put the sequence of characters between quotes. The quotes will not be sent to MIDI OUT port)

NOTE: all field must separated by an empty space!

It is possible to mix the three formats in the same cue definition. For example, the following sequence is valid: **$90 70 90 $C0 12 27** (this sequence sends a Note On, note 70, velocity 90, followed by a Control Change for control 12 with a value of 27). When cue data is programmed, the module will check the message validity. If the message is not valid, the cue is displayed in red and is not recorded in the module's cue memory.

Note that MIDI Running Status shall not be used in a cue. All MIDI messages defined in a cue shall contain the complete MIDI message.

4.1.3.3 - Controlling MIDI output stream from host

The EZ-CC MIDI sends the MIDI commands stored in the programmed cue, but it also forwards any MIDI message coming from the host. You can then prepare MIDI messages in Max and sends them dynamically, without needing to prepare a cue.

The EZ-CC MIDI can not receive Sysex messages from the host. These messages shall be stored as cues in the module.

All other MIDI messages coming from the patch are forwarded to MIDI OUT port, without being affected by the module. Note that Program Change messages can be filtered out if the "Use Program Change for Cues" check-box is checked. If you want to send a Program Change message from host when this selection is done, you have to program it in the cue.

4.1.3.4 - Sending MIDI input stream to host

The EZ-CC MIDI module is also able to process incoming MIDI data from the Kiss-Box. All MIDI data received on the MIDI IN port (excepted System Exclusive) will be forwarded directly to the patch, without being modified by the module.

Note that System Exclusive messages are not transmitted to host. If you need to receive and process System Exclusive messages, you can use the QConductor/M Manager module family.

4.1.4 - EZ-CC Serial4

The EZ-CC Serial4 is designed to control a serial port from a COM4 Kiss-Box.

Messages sent on serial port are stored as cues (built using the module GUI). They can also be built dynamically from the host using MIDI CC.

This module is able to process incoming serial data and send them to host.

4.1.4.1 - Configuration process

Each EZ-CC Serial4 module uses a default configuration when it is instantiated in the host. You need to change this configuration before the EZ-CC is able to control the remote Kiss-Box.

The top leftmost edit box allows the user to enter a name for the module, in order to help the programmer to remember what Kiss-Box is controlled by the module (e.g: "Video Projector 4").

Enter in *Device IP* edit box the IP address of the MIDI Kiss-Box you want to control from this module. The *Device Port* edition box shall be filled with the UDP port number declared during the Kiss-Box configuration process (see Kiss-Box Editor User's Manual). The *Local Port* defines the computer port on which the COM4 will send incoming serial data.

Then choose the serial port (1 to 4) to which serial data will be sent on the COM4 Kiss-Box.

NOTE: the complete module configuration is stored in the cue bank. You need to save the bank in order to save the configuration parameters (see section 3.1.2.2).

4.1.4.2 - Programming cues

The EZ-CC Serial4 module stores 128 cues. Each of these cues contains an ASCII string that will be transmitted to the serial port when the cue is activated. The maximum size of a MIDI message set is 64 bytes.

Each cue can be renamed using the edition box on the GUI window.

Note that cue parameters are saved automatically if a new cue is called. You can then switch from a cue to another without needing to save the changes each time.

The serial message can be entered in the cue editor using three different formats:

- hexadecimal (enter the ASCII code in hexadecimal with a "$" in front of the code)
- decimal (enter the ASCII code directly in the string)
- an ASCII string (put the sequence of characters between quotes. The quotes will not be sent to the serial port)

NOTE: all fields must separated by an empty space!

It is possible to mix the three formats in the same cue definition. When cue data is programmed, the module will check the message validity. If the message is not valid, the cue is displayed in red and is not recorded in the module's cue memory.

4.1.4.3 - Creating and sending strings from host

The EZ-CC Serial4 sends the ASCII string stored in the programmed cue, but the host is also able to create dynamically ASCII strings, by sending MIDI commands to the module.

Note: when a string is created from the host, it replaces the string coming from the cue memory, but only in the sending buffer. The cue memory itself is not affected. If a Program Change code is received, the string buffer content will be loaded with the original cue string.

Following MIDI commands are available to build dynamic strings from host:

- CC0 : clears the string buffer
- CC1 : appends a byte to the current message (byte value 0..127)
- CC2 : appends a byte to the current message (byte value 128..255)
- CC3 : sends a request to retrieve incoming serial data
- Note On (any) : sends the ASCII string to serial port
- Program Change : loads the ASCII string coming from a cue in the string buffer (the string is not sent)

5 - Document revisions

Date	Auteur	Version	Description
20/02/2010	B.Bouchez	0.1	First draft
28/03/2010	B.Bouchez	0.2	Screenshots remade on Mac platform Typo errors corrected Information added on MSP startup on Mac platform
29/03/2010	B.Bouchez	0.3	Clarification added to cue string definition for KBSerial4EZCC and KBMIDIEZCC
09/05/2010	B.Bouchez	1.0	Complete document review after changes have been integrated in the first release version of the EZ-CC modules